



PUC Minas

# Linguagens de Programação (Ciência da Computação)

Professora: M. Sc. Luciana De Nardin  
luciana@pucpcaldas.br

# 1

## Tipos de dados

## ◎ Tipos de dados

- ◎ Primitivos
- ◎ Cadeia de caracteres
- ◎ Ordinais definidos pelo usuário
- ◎ *Arrays*
- ◎ *Arrays* associativas
- ◎ Registro
- ◎ União
- ◎ Ponteiro
- ◎ Conjunto

## ◎ Tipos primitivos

- ◎ Tipos não definidos em função de outros tipos

- Ex.: *int, float, char, boolean*



## ◎ Tipos cadeia de caracteres

### ◎ Sequências de caracteres



Em C, as cadeias de caracteres devem ter um tamanho estático ou podem ter um tamanho dinâmico?

O que acontece quando uma cadeia maior é atribuída a uma cadeia menor ou vice-versa?

### ◎ Operações em C (*strcat*, *strcpy*, *strcmp*, *strcmpi*)

■ `char *str = "apples";` é válida?

## ◎ Tipos ordinais definidos pelo usuário

- ◎ Aquele cuja faixa de valores possíveis pode ser facilmente associada ao conjunto de números inteiros positivos

- Ex.: enumerações

```
Type DIAS is (Seg, Ter, Quar, Qui, Sex, Sab, Dom); // Pascal  
Enum dias {seg, ter, qua, qui, sex, sab, dom}; // C++
```

## Tipos ordinais definidos pelo usuário

- Em Pascal, não é permitido que um literal constante seja usado em mais de uma definição de tipo de enumeração em um determinado ambiente de referenciamento

```
type COR = (azul, laranja, marrom);  
type FRUTA = (maca, pera, laranja);
```



**Código inválido!!!**

## ⦿ Tipos ordinais definidos pelo usuário

### ⦿ Ex.: enumeração em C

```
enum dias_semana{dom, seg, ter, qua, qui, sex, sab};  
  
int main()  
{  
    int sem;  
    for (sem = dom; sem <= sab; sem++)  
        printf (" %d ",sem);  
  
    return 0;  
}
```

**Resultado:**

**0 1 2 3 4 5 6**



## ◎ **Tipos *arrays***

- ◎ É um agregado homogêneo de elementos de dados cujo elemento individual é identificado por sua posição no agregado em relação ao primeiro
- ◎ São referenciados por
  - 1º) Nome do agregado
  - 2º) Seletor dinâmico que consiste em ou mais itens conhecidos como subscritos ou índices

## ◎ Tipos *arrays*

### ◎ Envolve dois tipos distintos

- Tipo do elemento
- Tipo do(s) sobrescrito(s): geralmente inteiros, mas PASCAL e ADA permitem booleano, caractere e enumeração

### ◎ Limite interior: implícito

- C, C++ e Java: 0
- FORTRAN I, II e IV: 1
- Outras: especificadas pelo programador

## ◎ Tipos *arrays*

### ◎ Operações com *arrays*

- São parecidas com as das strings >> depende da linguagem
- Como se soma duas matrizes em C?

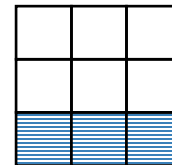
### ◎ Fatias (*slices*) em FORTRAN

INTEGER VETOR(1:10), MAT (1:3,1:3), CUBO (1:3,1:3,1:4)

VETOR(3:6) → do terceiro ao sexto elementos



MAT(3,1:3) → Terceira linha da matriz



## ◎ **Arrays associativos**

- ◎ Suportados pelo padrão Java (PHP e Python)
- ◎ É um conjunto não ordenado de elementos indexados por um número igual de valores chamados chaves
- ◎ Cada elemento de um *array* associativo é um par de entidades: **chave** (nome para o elemento) e um **valor**
- ◎ Também chamados de *hashes*

## ◎ Arrays associativos

### ◎ Exemplo em Perl

```
%salarios = ("Cedric" => 75000, "Perry" => 57000, "Mary" => 55750,  
"Gary" => 47850);
```

```
$salarios{"John"} = 58850; // adiciona um item
```

```
delete $salarios{"Gary"}; // remove um item
```

```
@salarios = (); // esvazia o hash
```

```
if (exists $salarios{"Shelly"}) .... // retorna V ou F
```



## ○ Tipo registro

- É um agregado possivelmente homogêneo de elementos de dados



Qual a diferença entre um registro e um *array*?

*Array* →  
homogeneidade de  
elementos e referência  
através de índices

Registro →  
heterogeneidade de  
elementos e referência  
por identificadores

## ◎ Tipo registro

### ◎ Exemplo em COBOL

```
01 REGISTRO-EMPREGADO.  
  02 NOME-EMPREGADO.  
    05 PRIMEIRO      PICTURE IS X(20).  
    05 MEIO          PICTURE IS X(10).  
    05 ULTIMO        PICTURE IS X(20).  
  02 TAXA-HORARIA    PICTURE IS 99V99.
```

01, 02 e 05 são números de nível → estrutura hierárquica  
X(20) → 20 caracteres alfanuméricos  
99V99 → 4 dígitos decimais com a vírgula no meio

**Acesso** → MEIO OF NOME-EMPREGADO OF REGISTRO-EMPREGADO

## ◎ Tipo registro

### ◎ Exemplo em ADA

```
REGISTRO_EMPREGADO: record
  NOME_EMPREGADO: record
    PRIMEIRO: STRING (1..20);
    MEIO: STRING (1..10);
    ULTIMO: STRING (1..20);
  end record;
  TAXA_HORARIA: FLOAT;
end record;
```

**Acesso** → REGISTRO\_EMPREGADO.NOME\_EMPREGADO.MEIO

## ◎ Tipo união

- ◎ Pode armazenar diferentes valores de tipo durante a execução do programa

## ◎ Exemplo em C

```
union aluno
{
    char nome[10];
    int RA;
} id;
```

## ◎ Tipo conjunto

- ◎ São aquelas cujas variáveis podem armazenar coleções não-ordenadas de valores distintos de algum tipo ordinal chamado de **tipo-básico**

## ◎ Exemplo em Pascal

```
if (ch = 'a') or (ch = 'e') or (ch = 'i') or (ch = 'o') or (ch = 'u')...  
  
if ch in ['a','e','i','o','u']...
```



## ◎ Tipo conjunto

### ◎ Operações sobre conjuntos

- **União** → a união de dois conjuntos A e B é conjunto de todos os objetos que são elementos de A ou B ou de ambos
- **Interseção** → a interseção de dois conjuntos A e B é o conjunto de todos os objetos que são elementos de A e B simultaneamente
- **Diferença** → a diferença de dois conjuntos A e B é o conjunto de todos os objetos que são elementos de A mas, não de B

## ◎ Tipo conjunto

### ◎ Operações sobre conjuntos

- **Teste de inclusão** → um conjunto A está incluso em um conjunto B (A é “subconjunto” de B) se, e somente se, todo elemento de A é também um elemento de B e vice-versa
- **Teste de presença** → um dado objeto pode ser ou não um elemento de um dado conjunto

## ◎ Tipo conjunto

### ◎ Exemplo em Pascal

- Fazer um programa em PASCAL que leia uma frase de até 30 caracteres, determine e escreva o número de vogais, consoantes e espaços em branco existentes na frase.

```
program Pesquisa;  
  type Alfabeto = 'A' .. 'Z' ;  
    Letras = set of Alfabeto;  
var I, {Subscrito}  
    Vog,    {Contador de vogais}  
    Cons,   {Contador de consoantes}  
    Brancos: {Contador de espaços em branco} integer;  
  
    Frase: string [30]; {Sequencia de caracteres}  
  
    Vogais, {Conjunto de vogais}  
    Consoantes: Letras; {Conjunto de consoantes}
```

**begin**

```
Vogais := ['A', 'E', 'I', 'O', 'U'];  
Consoantes := ['A' .. 'Z'] - Vogais;  
Vog := 0;  
Cons := 0;  
Branco := 0;  
readLn (Frase);  
for I := 1 to 30 do  
    if Frase [I] = ' ' then  
        Branco := Branco + 1  
    else if Frase [I] in Vogais then  
        Vog := Vog + 1  
    else if Frase [I] in Consoantes then  
        Cons := Cons + 1;  
writeln (Vog, Cons, Branco);
```

**end.**



## ◎ Tipo ponteiro

- ◎ Um ponteiro é uma variável cujo conteúdo é um **endereço de memória**
- ◎ Um tipo ponteiro é aquele em que as variáveis possuem uma faixa de valores que consiste em endereços de memória e um valor especial, *nil*
- ◎ O *nil* não é um endereço válido e serve para indicar que um ponteiro não pode ser usado para referenciar qualquer célula da memória

## ◎ Tipo ponteiro

- ◎ Foi criado para:
  - Endereçamento direto (*assembly*)
  - Gerenciamento de armazenamento dinâmico (*heap*)
- ◎ Ponteiros não são tipos estruturados (como *arrays* e registros são) ainda que sejam definidos usando-se um operador de tipo

## ◎ Bibliografia

SEBESTA, R. W. , **Conceitos de Linguagens de Programação – 5ª**  
Edição, Porto Alegre: Bookman, 2003.